

CONTROLLER FOR AUTOMOBILE

Patent Number: JP7277105
Publication date: 1995-10-24
Inventor(s): KURATA KENICHIRO; others: 04
Applicant(s): HITACHI LTD
Requested Patent: ☐ JP7277105
Application Number: JP19940076845 19940415
Priority Number(s):
IPC Classification: B60R16/02; G05B15/02; G06F9/06
EC Classification:
Equivalents:

Abstract

PURPOSE: To reduce development man-hours for control software and to provide a general purpose applicability in relation to a hardware change.

CONSTITUTION: Automobile control software to be described is constructed of separate components consisting of an application part 1, an I/O description part 2, and an I/O data part 3, and these are connected to each other by means of a software connecting means 4. In this way, description is carried out while the application part 1 and the I/O description part 2 are separated from each other, so that description software is simplified. As the I/O data part 3 is arranged independently of the I/O description part 2, alteration of hardware can be accomplished only by alteration of the I/O data part 3. In addition, an object program is generated by connecting these parts, and therefore the optimization process can be programmed.

Data supplied from the **esp@cenet** database - l2

本制御ソフトウェア構成、およびそれらの結合処理方法を提供するための図である。

[0008]

図面を解決するための手段] 前記目的を達成するため、本発明では、制御ソフトウェアに、制御内容を記述したアプリケーション部と、ハードウェアに対する入出力処理について記述したI/O記述部とを設け、これらI/O記述部3に示されたハードウェアに関する情報と、結合手段を用いて結合し、目的のプログラムを得るようにした。

[0009]

[作用] このように構成され、如図に示される本発明によれば、複雑な自動車制御ソフトウェアの作成時にも、アプリケーション部とI/O記述部2を分けたために、ハードウェアに対する入出力動作に関するソフトウェア設計と、制御のための処理動作に関する設計とを別々にすることができ、さらに、それぞれが別々に設計されていることから、アプリケーションや入出力処理が変更された場合にも記述の一部を修正し、結合することによって、容易に目的のプログラムを得ることができる。

[0010] また、ハードウェアに対する入出力処理内容をI/O記述部2に記述された表現を用いて記述し、ハードウェア自身に関するデータ1/Oデータ部3にそれぞれ分離して記述するようにしたこと、ハードウェアの変更時でも、ハードウェアに付随してI/Oデータ部3を差し替えればその他の部分はそのまま使用する事ができる。

[0011] さらに、別々に作った各部を、結合手段によって結合して目的のプログラムを生成させる方式としたことから、結合手段にプログラムの最適化処理のノウハウを記述させることができる。

[0012]

[実施例] 以下、本発明の実施例を図面に基き詳細に説明する。

[0013] 図1は本発明の第1実施例である。図において、ソフトウェア開発時に各プログラム作成者、またはソフトウェア開発者によって記述された記述されたソフトウェア6には、アプリケーション部1、I/O記述部2、I/Oデータ部3が設けられている。このアプリケーション部1には、制御において、制御対象の動作内容、およびそのための各種を決定するための処理や最適化処理などの、制御の本質的な内容が記述される。また、I/O記述部2には前述のアプリケーション部1で必要とされるハードウェアの各種または属性をソフトウェア上に取り込み、あるいはアプリケーション部1において、制御対象の動作内容や動作内容をハードウェアへ出力する各種処理などの、ハードウェアに対する入出力処理、およびその最適化処理が一般化された表現で記述されている。さらにI/Oデータ部3には、ハードウェアの構成や機能、その設定方法、アドレス

ステータなどの一連の属性データなど、従来ハードウェアマニュアルに記述されていた事項がアプリケーション部1やI/O記述部2で使用するし、またこれらに記述される他に、制御対象とマイコン端子の接続など、システム全体の構成に関する事項や、目的の制御に特有のハードウェア属性を容易に実現するためのデータ、ハードウェアとソフトウェア間の使用に関する取り決めから自動的に決まるデータなどが記述されており、各制御内容の記述を容易にし、従来の事項についての記述を省略可能とする事で、プログラムの記述を容易にしている。

なお、これらのデータは後述するソフトウェア結合手段5において読み出され、利用できるように整理された形で記述されている。

[0014] これらの記述ソフトウェアの各部分は、記述後、ソフトウェア結合手段4によって結合され、ハードウェア上で実行される制御動作を行う目的のプログラム5が生成される。このとき、ソフトウェア結合手段4は、そのソフトウェアを格納するハードウェアに付随した各データ1/Oデータ部3に記述されたデータを参照しながらI/O記述部2を参照し、アプリケーション部1とのリンクを行うことによって目的のプログラムを生成させるとともに、プログラムの最適化を行い、無駄な処理を取り除き、効率の良い目的のプログラム5を作り出す。

[0015] ここで、このソフトウェア結合手段4は、前述のアプリケーション部1、あるいはI/O記述部2、あるいはI/Oデータ部3に含まれていても構わない。あるいはこれら3つとは別々に設けられていても構わない。

[0016] 前述のように、従来ハードウェアマニユアルの形で記述されていた、マイコン等のハードウェアに関する情報も、ユーザの使用形態に合わせて整理し、データベース化した際のユーザインターフェース型ハードウェアマニユアルとして提供すること、ユーザは独自の表現、および方法でソフトウェアの開発をする事が可能となり、同時に、ハードウェアの変更を行った場合でも、付属のユーザインターフェース型ハードウェアマニユアルを用いることで、きわめて少ない作業工数でソフトウェアをそのまま利用することが可能となる。

[0017] 図2は本発明の第2実施例である。図において、記述ソフトウェア25には制御の本質的な内容が記述されたアプリケーション部21、ハードウェアに対する入出力動作のほかにハードウェアの構成、機能、使用法など、入出力処理およびハードウェア関連データの記述されたI/O記述部22が設けられている。

[0018] これらは、ソフトウェア結合手段24によって結合され、最適化されてハードウェア上で実行される制御動作を行う目的のプログラムが生成される。ただし、このソフトウェア結合手段は、前述のアプリケーション部21、あるいはソフトウェア記述部22に含まれてもよい。

いし、あるいはこれら2つとは独立して記述されている場合もある。

[0019] I/Oソフトウェア部22には、ユーザの希望するI/O動作仕様またはアプリケーション部1に記述された入出力動作の命令に応じて、使用ハードウェアに適合した入出力動作を行うようなソフトウェアが記述されるため、ハードウェアの変更に応じて、変更後のハードウェアに対応したI/O記述部22に、変更後の使用条件を考慮してI/O記述部221はほとんど変更せずに使用できるため、ソフトウェアの汎用性が図られる。

[0020] 図3は本発明の第3実施例である。図において、プログラム部31には制御の本質的な内容を記述したアプリケーション部、および、ハードウェアに対する入出力動作を一括化した表現で記述したI/O記述部、および、ハードウェアの構成、機能、使用法など、ハードウェアおよびその利用法に関するデータが記述されたソフトウェア結合手段5において利用できるように整理されて記述されている。

[0021] ソフトウェア結合手段33は前記I/Oデータ部32に記述されたハードウェアに関するデータや利用法、プログラムの最適化に関する申し送り事項などを参照しながらプログラムの最適化を行い、最適化処理を行って、ハードウェア上で実行される制御動作を行う目的のプログラム34を生成する。

[0022] ここで、このソフトウェア結合手段は、前述のプログラム部31、あるいはI/Oデータ部32に含まれていても構わない。あるいはこれら2つとは独立して設けられていても構わない。

[0023] 図4は前記図1で示した第1実施例について、ソフトウェア結合手段4.5が生成されるものである。図に示されるように、各ソフトウェア作成者によって記述されたアプリケーション部4.1、I/O記述部4.2、I/Oデータ部4.3は、ソフトウェア結合手段4.4およびソフトウェア結合手段4.5によって結合されるが、その順序は以下に示すとおりである。

[0024] まず第一に、アプリケーション部4.1とI/O記述部4.2がソフトウェア結合手段4.4によって結合され、プログラム部4.5が生成される。このとき、結合手段4.4のソフトウェア結合手段4.5は、制御動作とハードウェアに対する入出力動作のそれぞれについて動作タイミング、および処理動作の制御が行われ、目的の制御動作に適合したプログラム部4.5が生成される。次に、ソフトウェア結合手段4.4は、I/Oデータ部4.5を参照しながら、先に生成されたプログラム部4.5のI/O処理動作内容の記述を使用ハードウェアの仕様に合わせた書き方に変換し、目的のプログラム4.7

を得る。

[0025] 図6は前記図1で示した第1実施例について、ソフトウェアの結合方法、およびハードウェアへの搭載方法についての一実施例を示したものである。図に示されるように、記述ソフトウェア6にはアプリケーション部6.1、およびI/O記述部6.2が設けられ、これらは結合手段6.4によって結合され、生成プログラム6.5が生成される。また、I/Oデータ部6.3はROM6.7に記憶された形で提供され、制御装置上6.6の記憶生成プログラム6.5が必要に応じて呼び出して使用する。

[0026] ただし、生成プログラム6.5の持つI/Oデータ6.3利用機能については、アプリケーション部6.1、またはI/O記述部6.2にあらかじめ含まれていても構わない。あるいはソフトウェア結合手段6.5によって結合時に付加されてもよい。前記図1から図6において示してきたソフトウェア構成および結合方法を用いることによって、ソフトウェアの記述が簡便になることに加えて、各制御アプリケーション部およびI/O記述部をそれぞれ別々に設計し、結合することが可能となるので、多数の制御項目をもつ複雑な制御アルゴリズムの記述が多くなり、また、I/Oデータを格納することによって、共通のハードウェアをさまざまな用途に、かつ使いやすいう形で提供することが可能となる。

[0027] 例えはエンジン制御、AT制御のような複雑な制御ソフトウェアも、いくつもの小さなブロックに分けて記述することにより、極めて簡単に制御内容を記述できる。その一例として以下に簡単なエンジンの点火制御モデルの記述を示す。

[0028] 図7は説明するエンジン制御モデルのハードウェアシステム構成を示したものである。

[0029] エンジンの点火は図に示されたプラグ1〜6のギヤブ間電圧によって行うが、そのためには、コイル7.3に一定時間電圧を印加した後、電圧を切断する制御が行われ、電圧が切断された瞬間、コイルに誘えられた誘導電圧のエネルギーがキャップ間の放電の形で放出される。したがって、コイル7.3の電圧がoffしてからoffするまでの時間（以下遅延時間）の制御は、点火タイミング、すなわち電圧がoffするタイミングの制御と共に極めて重要なものである。

[0030] なお、本モデルにおいて、制御用のマイコン7.1にはエンジンのクランク角の送受信位置を知らせるためのRe f信号7.6、クランク角の回転数を知らせるP OS信号7.7、および吸入空気温度T w 7.8、冷却水温度T w 7.9などが入力されており、Re f信号7.6は120度おきに、P OS信号7.7は1度おきに入力される。また、IGN信号出力端子7.5からはプラグ7.4への制御信号が出力されるが、この出力においては、約120度おきに点火制御が行われる。実際にはエンジンは2回転に1回だけ点火すればよいので、この出力信号は分

配図72によって6つに分配され、6個のエンジンでは、各気筒がちょうど720度1回転点火するようになっている。

[0031] 図8はマイコン86内のハードウェア構成を示したもので、本モジュールで使用するタイマー89とCPU85が示されている。タイマー89にはタイマーカウンタ81と比較器82、コンパリアスタ83、タイマーコントロールレジスタ84があり、タイマーカウンタ81には前述のPOS信号が入力され、カウンタ角が1回転するたびにカウンタの値も1ずつ増えるようになっている。コンパリアスタ83の値とタイマーカウンタ81の値は常に比較器82によって比較され、2つが一致した時にはタイマー出力87を発生させる。CPU85には割り込み出力信号88を発生する事ができ、したがってタイマーを使って点火制御を行うには、タイマーの機能を選択するためのタイマーコントロールレジスタ84に必要な設定となるように値を書き込むとともに、動作を起こした時点のタイマーカウンタの値を調査し、CPUからコンパリアスタ83に書き込む。

[0032] なお、Ref値はCPU85に割り込みの形で入力される。

[0033] 図9はエンジン点火制御のタイムチャートを示したものである。制御したい内容は、「Ref値」の立ち上がり97からaだけ過ぎた時点Y96でICN信号を「高」(点火)する。また、「ICN」の遅延時間をbにする。事であるが、後者は前述したくないので「点火」時点Y96からbだけ手前の時点X96で出力をonにする」と記述することになる。

[0034] ここで、前述のコンパリアスタには動作を起こしたい時点のカウンタ値を代入しておくことになってるので、たとえば「Ref値」の立ち上がり97からaだけ過ぎた時点Y96」をセットする際には、まずRef値立ち上がり97の時のカウンタ値Y94を調べ、そこからaだけ過ぎた時点Y96を求め、コンパリアスタに代入する値はs+aである。

[0035] 図10に本モジュールを前記図1に示したソフトウェア構成により記述した一実施例を示す。図において、統合前のソフトウェアはアプリケーション部10から構成され、これは前記図4に示したように、まずアプリケーション部10と1/O部102が結合され、ついで、得られたプログラムを1/Oデータ部103を参照しながら実行するという手順で目的のプログラムを得る。

[0036] ここで、各部の内容について説明する。アプリケーション部10においては、その動作タイミングにおいて、処理を行うことを示す。したがってアプリケーション部10の内容は、まず、Ref信号が立ち上がった時点(Ref値の立ち上がり)に、変数X=

前記図3に示したソフトウェア構成によって記述したもので、同様に図12の1/Oデータを用い、図10例には前記図13に示した目的プログラムを得るが、前記図10の場合と比べて、ソフトウェア結合のみの処理が1段階減っているのが特徴である。

[0045] 図14は本発明のソフトウェア構成を自動専用シングルチップマイコンに適用した場合の一実施例である。ソフトウェア構成としては、アプリケーション部11、1/Oソフトウェア部22及び自動専用OS (operating system) 部101の3つに分けられる。1/Oソフトウェア部22は前述のように1/O部102及びCPU04 (中央演算装置) 部に属する1/Oデータ部3をソフトウェア結合手段104に入力し作成する。そして、上記1/Oソフトウェア部22及び自動専用OS部101をシングルチップマイコン102に記憶手段103 (ROM: 読み出し専用メモリ、EPROM: 電気的に書き換え可能なメモリ) に記憶させユーザーに提供する。ユーザーはアプリケーション部11を作成し、上記記憶手段103に書き込み動作を実行する。これにより、例えば、シングルチップマイコン102の性能及びハード構成が変更になった場合、更に、1/O部102が変更になった場合でも、ユーザーはアプリケーション部11を移植することができ、アプリケーション部11を移植し、マイコンカーは1/Oデータ部3を変更して自動専用ソフトウェア付シングルチップマイコン102を提供すれば良い。また、シングルチップマイコン102を置き換えるだけで、アプリケーションソフトを作成することができる。

[0046] 図15はソフトウェア構成を自動専用OSプログラムとした場合の一実施例である。ここでは、上記1/Oソフトウェア部22がタスク管理及び優先順位等から成るタスクディスパッチの逐次OS105を用いて自動専用OS106を作成する。このOSは自動車専用であり、自動車制御として汎用性もある。また、図14と同様にシングルチップマイコン102の記憶手段103に書き込んで提供することも可能である。

[0047] [発明の効果] 本発明によれば、アプリケーション部と1/O部102とを分けて記述するようにしたために、ハードウェアに対する出力動作に関するソフトウェア設計と、制御のための演算処理動作に関する設計とを別々に行うことができる。プログラムのソフトウェア設計が煩雑になり、プログラムの作成工数が削減する共に、プログラムの信頼性を上げることができる。

[0048] また、目的のプログラムを得るのに、アプリケーション部と1/O部102とを分けて記述し、それらを結合する方法を用いるために、アプリケーションや出力処理が変更された場合でも記述の一部を修正し、結合することによって、容易に目的のプログラムを得ることができるので、ソフトウェア汎用性が高くなる。さ

らに、ハードウェアに対する出力処理内容を1/O部102に記述した表を用いて記述し、ハードウェアに関するデータを1/Oデータ部3にそれぞれ分離して記述するようにしたことから、ハードウェアの変更時、ハードウェアに付随して1/Oデータ部3を設計すれば、それ以外の部分はそのまま使用する事ができる。ハードウェアソフトウェアの汎用性が高くなる。

[0049] さらに、別々に作った各部を、結合手段によって結合して目的のプログラムを生成させる方式として、たとえば、結合手段にプログラムの高度化処理のノウハウを記述させることができるので、プログラム作成は特に要することなく、プログラム作成技術の継承が行える。

[図面の簡単な説明]

[図1] 本発明による、アプリケーション部、1/O部102、1/Oデータ部3に分けて記述されたソフトウェアを結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

[図2] 本発明による、アプリケーション部、1/O部102、1/Oデータ部3に分けて記述されたソフトウェアを結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

[図3] 本発明による、アプリケーション部、1/Oデータ部3に分けて記述されたソフトウェアを結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

[図4] 本発明による、別々に記述されたアプリケーション部、1/O部102とを結合し、さらに1/Oデータ部3を結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

[図5] 本発明による、別々に記述された1/O部102、1/Oデータ部3を結合し、さらにアプリケーション部と結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

[図6] 本発明による、アプリケーション部、1/O部102とを結合し、さらに1/Oデータ部3をデータ用ROMに載せ、データを参照しながらプログラムを動作させることを示した構成図。

[図7] エンジンの点火制御を行うに必要なハードウェア構成の説明図。

[図8] タイマーを含むマイコンの動作を説明するための構成図。

[図9] 点火制御モデルの動作を説明するためのタイムチャート。

[図10] 点火制御をアプリケーション部、1/O部102、1/Oデータ部3に分けて記述した例を示すプログラム図。

[図11] 点火制御をプログラム部、1/Oデータ部3に分けて記述した例を示すプログラム図。

11

【図12】ハードウェアデータを結合手段が利用できる形に整理して示したI/Oデータテーブル図。

【図13】本発明による分岐処理を行ったソフトウェアが、結合処理によって一連のソースプログラムとなることを示したプログラム図。

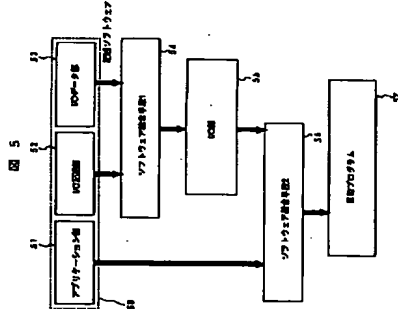
【図14】本発明のソフトウェア構成を自動車用シングルチップマイコンに展開した場合の一実施例を示す図。

12

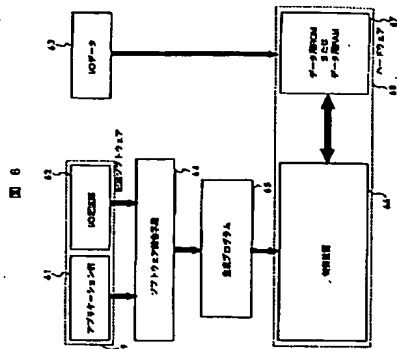
【図15】本発明のソフトウェア構成を自動車用OSプログラムとした場合の一実施例を示す図。

【符号の説明】
1...アプリケーション部、2...I/O配速部、3...I/Oデータ部、4...ソフトウェア結合手段、5...目的プログラム。

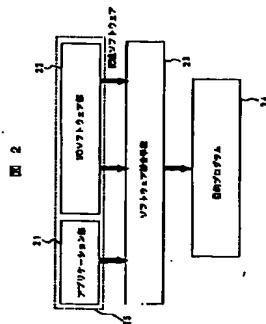
【図5】



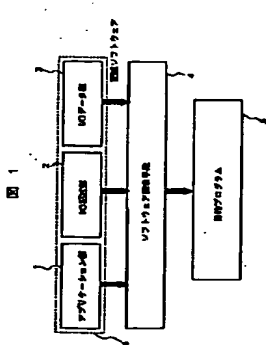
【図6】



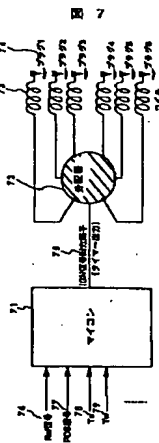
【図2】



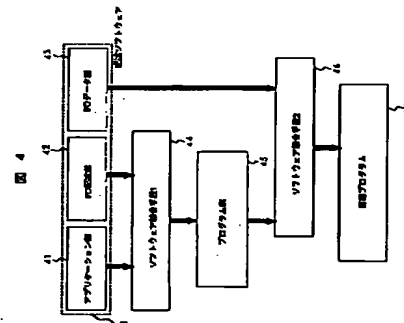
【図1】



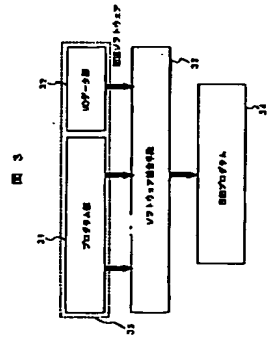
【図7】



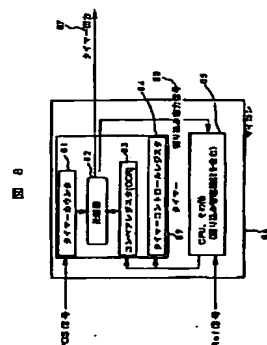
【図4】



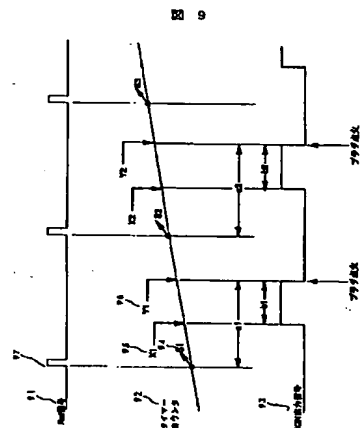
【図3】



【図8】



【図9】



【図13】

図 13

(a) Initializationへ要略

```

char DUMMY;
Init_Ign()
{
    IGN_TCR = IGN_TCR_INIT_DATA;
    IGN_TCSR = IGN_TCSR_INIT_DATA;
    SYSCAL = SYSCAL_INIT_DATA;
    SYSCAL = SYSCAL_INIT_DATA;
    PLDRA = PLDRA_INIT_DATA;
    PDRR = PDRR_INIT_DATA;
    IPRA = IPRA_INIT_DATA;
    IPRD = IPRD_INIT_DATA;
}

```

(b) Background Jobへ要略

```

On_Ign()
{
    b = k * (1/No);
    a = f(No, Tc, ...);
}

```

(c) Ref Jobへ要略

```

Ref_On_Ign()
{
    register char s = IGN_CNT;
    No = a-b;
    x = a;
    #pragma asm
    STC.W SR, 0-SP
    LDC.W #5215, SR
    #pragma endasm
    if (x > IGN_CNT)
    {
        IGN_OCLA = x;
        IGN_TCR = IGN_TCR_ON_DATA;
        IGN_TCSR = IGN_TCSR_ON_DATA;
    }
    #pragma asm
    LDC.W #5215, SR
    #pragma endasm
}

```

フロントページの続き

(72)発明者 石井 潤市

茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 森永 茂樹

茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内